UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/735,592	12/12/2000	Michael Wayne Brown	AUS9-2000-0720-US1	8578

35525 7590 03/07/2006

IBM CORP (YA)
C/O YEE & ASSOCIATES PC
P.O. BOX 802333
DALLAS, TX 75380

EXAMINER	
ALI, SYED J	
ART UNIT	**PAPER NUMBER**
2195	

DATE MAILED: 03/07/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/735,592	BROWN ET AL.	
	Examiner	Art Unit	
	Syed J. Ali	2195	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *16 December 2005*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-22* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-22* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This office action is in response to the appeal brief filed December 16, 2005. Claims 1-22 are presented for examination.

2. The text of those sections of Title 35, U.S. code not included in this office action can be found in a prior office action.

Claim Rejections - 35 USC § 101

3. **Claims 1-22 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.**

4. Claims 1-22 are non-statutory by virtue of failing to produce a "useful, concrete, and tangible result." *State Street Bank & Trust Co. v. Signature Financial Group Inc.*, 149 F. 3d 1368, 1373-74 (Fed. Cir. 1998). The claims are directed to a method of executing a program asynchronously in distinct threads, but fail to indicate how this is method of execution accomplishes a practical application. That is, the claims represent nothing more than an idea or concept, i.e. an algorithm, which is ineligible for patent protection. *Diamond v. Diehr*, 450 U.S. 175, 185 (1981).

5. While such an idea is ineligible subject matter in itself, a practical application of such an idea may be statutory subject matter. For instance, Applicant's specification indicates that asynchronous execution of applications in a multithreaded environment may be useful for speeding up the processing thereof (pg. 2). Allowing the concurrency to be specified in software

rather than hardware gives developers more flexibility to realize the concurrency potential of their applications (pg. 2). However, the claims remain non-statutory because they are devoid of any such practical applications and merely recite the algorithm.

Claim Rejections - 35 USC § 103

6. **Claims 1-22 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bonachea ("Bulk File I/O Extensions to Java").**

7. As per claims 1-4, Bonachea teaches the invention as claimed, including a method for asynchronous execution within a program, comprising:

executing code in a first thread (§ 4.2);

determining whether a first keyword exists in the code, the first keyword being a flag indicating that a subsequent code element following the first keyword may be executed out of order (§§ 4.1.1 - 4.1.3);

executing the code element in a second thread (§ 4.1.2), wherein the first thread is executed on a first processor and the second thread is executed on a second processor (§ 2.2); and

wherein the code element is one of an instruction, block, and a method (Fig. 2) and the first keyword is usable in both an internal definition of a method and a type definition for the method (Figs. 2-3 and 5, wherein the AsyncFile extension is designed as a class, which can be used as an object type or a return type).

8. While Bonachea does not specifically recite spawning a new thread in response to encountering a keyword during the executing of the code, it would have been obvious to one of ordinary skill in the art that such is well within the bounds of what is disclosed by Bonachea. Bonachea implements a library that supports asynchronous I/O for Java, but tested the results using a dialect of Java without a JVM. There is nothing to suggest that the extensions could not be used in Java; the library is designed with such a use in mind. As is well-known in the art, there are several ways that Java code can be executed, including by interpretation, just-in-time compiling, or fully compiling code before it is executed. If interpretation was chosen, the creation of a new thread would occur at runtime; in the embodiment shown by Bonachea, the runtime data structures are created at compile time, which is nothing more than a design choice. Java easily supports each type of execution.

9. It is also noted that Bonachea does not support conventional multithreading, whereby asynchronous threads of control may be created (§ 2.2). However, this is a limitation of the Titanium compiler chosen by Bonachea to implement the asynchronous extensions, not of Java itself. In fact, the use of multiple threads of control is a well-known feature of Java, but may be limited by the underlying architecture that is chosen. Nonetheless, it is well-known that there are a myriad of scheduling choices that can be made with regard to how threads are executed, i.e. multiple threads on a single processor or one thread per processor, as claimed. Stanbach (US 2001/0049747) discusses how threads may be executed on the same or different processors, depending on the particular needs of the chosen implementation (paragraph 0046).

10. As per claims 5-6, Bonachea teaches the invention as claimed, including a method for asynchronous execution within a program, comprising:

executing code in a first thread (§ 4.2);

determining whether a first keyword exists in the code, the first keyword indicating a code element that may be executed out of order (§§ 4.1.1 - 4.1.3);

executing the code element in a second thread (§ 4.1.2);

determining whether a second keyword exists in the code, the second keyword indicating that execution of the code element in the second thread must complete before a next code element immediately following the second keyword is executed (§ 4.1.3, wherein the "Done" methods provide a means of synchronizing the asynchronous operations by requiring the I/O operations be complete before execution continues; that the "Done" method operates by polling is immaterial, as the claim limitation only requires the keyword require the asynchronous operation to complete before execution may progress);

executing the next code element in the first thread after execution of the code element in the second thread completes (§ 4.1.3);

determining whether a third keyword exists in the code element, the third keyword indicating a statement that may be executed out of order (§§ 4.1.1 - 4.1.3); and

executing the statement in a third thread (§ 4.1.2).

11. It is noted that Bonachea does not explicitly demonstrate an example of how nesting may be supported. However, the fact that this feature is not discussed does not mean it is not supported. Bonachea discusses the initiation of asynchronous I/O operations, which execute in the background while the foreground application continues. The background application may

require further I/O, in which case another asynchronous request may be initiated to complete the processing.

12. As per claims 7 and 9, Bonachea teaches the invention as claimed, including the method of claim 1, wherein the method is executed by an interpreter (§ 2.2) and the second thread is a lightweight thread (§ 4.1.2) (see paragraph 7 above).

13. As per claim 8, Bonachea does not teach the interpreter being a Java virtual machine.

14. Bonachea implements the Java extensions in Titanium, which is actually a dialect of Java and does not utilize a JVM (Abstract; § 2.2). Titanium was developed exclusively at the University of California at Berkeley and functions as a superset of Java. The language extensions defined by Bonachea are for the purpose of adding bulk array operations and enabling asynchronous I/O in Java. However, one of the concerns presented is the portability to other platforms besides Solaris, including those that implement Java within a virtual machine framework (§ 6). As such, it would have been obvious to one of ordinary skill in the art to use the same language constructs on top of a Java Virtual Machine, as this would allow portability between platforms and enable concurrent asynchronous non-blocking execution. The extensions are designed explicitly with Java in mind; it would have been obvious to implement the extensions in Java, as the majority of Java users employ a JVM, rather than the specialized dialects used by Bonachea.

15. As per claims 10-13 and 16, Bonachea teaches the invention as claimed, including an apparatus for performing the method of claims 1-4 and 9, respectively (§ 2.2).

16. As per claims 14-15, Bonachea teaches the invention as claimed, including an apparatus for performing the method of claims 5-6, respectively (§ 2.2).

17. As per claim 17, Bonachea teaches the invention as claimed, including an apparatus for asynchronous execution within a program, comprising:

an interpreter (§ 2.2); and

a program, the program including a first keyword indicating a code element that may be executed out of order (§§ 4.1.1 - 4.1.3), wherein upon detecting the keyword, a light weight thread is created (§ 4.1.2) and executes the code element in the light weight thread (§ 4.1.2).

18. As per claim 18, Bonachea teaches the invention as claimed, including an apparatus for performing the method of claim 8 (§ 2.2).

19. As per claims 19 and 20, Bonachea teaches the invention as claimed, including a computer program product, a computer readable medium for performing the method of claims 1 and 4, respectively (§ 2.2).

20. As per claims 21-22, Bonachea teaches the invention as claimed, including a computer program product, a computer readable medium for performing the method of claims 5-6, respectively (§ 2.2).

Response to Arguments

21. **Applicant's arguments with respect to the rejections of claims 10-18 under 35 U.S.C. § 101 have been fully considered and are persuasive.**

22. The claimed apparatus is tangibly embodied since they are "means-plus-function" claims and cover the corresponding structure in the specification. Therefore, the rejection has been withdrawn. However, upon further consideration, a new grounds of rejection is made in view of the claims failure to recite a "useful, concrete, and tangible result." See numbered paragraphs 3-5 above.

23. **Applicant's arguments with respect to the art rejections over Bonachea have been fully considered but they are not persuasive.**

24. Applicant alleges that Bonachea fails to teach a capability for determining at runtime whether to execute a code segment asynchronously. Applicant alleges that Bonachea cannot be modified to include a runtime interpreter since the specific embodiment utilizes a compiler. However, this argument specifically ignores that Bonachea explicitly states that any Java environment could be used, but that the specific results described are in relation to a compiler. That is, "the library interfaces are appropriate for standard Java and the implementation could be

easily ported. It is our belief that the concepts and performance results are relevant to any high-performance dialect of Java." (§ 1) (emphasis added).

Furthermore, Applicant's argument that the suggestion or motivation to modify the reference must be "in the reference." (see pg. 13 of Applicant's appeal brief). However, this is simply an incorrect statement of the law. There is no requirement that the suggestion come from the reference itself, but may also be drawn from the nature of the problem to be solved, the teachings of the prior art, or the knowledge of persons of ordinary skill in the art. *In re Rouffet*, 149 F.3d 1350, 1357 (Fed. Cir. 1998). It is well settled that the suggestion may be express or implied, and any allegation that the motivation must be in the reference carries no weight, being a narrow and erroneous statement of the law. Moreover, Applicant's argument is not persuasive because there is an explicit suggestion to modify Bonachea in the reference itself, as discussed above. Moreover, Bonachea provides detailed classes and libraries that can be used in Java; there is nothing to suggest that an interpreter could not be used, in which case the detection of keywords would have to occur at runtime. Bonachea's explicit suggestion that interpreters are well-suited to exploit the features described puts this matter to rest.

Conclusion

25. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Syed J. Ali whose telephone number is (571) 272-3769. The examiner can normally be reached on Mon-Fri 8-5:30, 2nd Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai T. An can be reached on (571) 272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Syed Ali
March 2, 2006